Transition Semantics
The Dynamics of Dependence Logic

Pietro Galliani
ILLC
University of Amsterdam
The Netherlands
(pgallian@science.uva.nl)

February 24, 2013

Abstract

We examine the relationship between Dependence Logic and game logics. A variant of Dynamic Game Logic, called *Transition Logic*, is developed, and van Benthem's representation theorem for First-Order Logic and Dynamic Game Logic is adapted to the case of Dependence Logic and Transition Logic.

This suggests a new perspective on the interpretation of Dependence Logic formulas, in terms of assertions about *reachability* in games of imperfect information against Nature. We then capitalize on this intuition by developing expressively equivalent variants of Dependence Logic in which this interpretation is taken to the foreground.

1 Introduction

1.1 Dependence Logic

Dependence Logic [16] is an extension of First-Order Logic which adds *dependence atoms* of the form $=(t_1 \ldots t_n)$ to it, with the intended interpretation of "the value of the term t_n is a function of the values of the terms $t_1 \ldots t_{n-1}$."

The introduction of such atoms is roughly equivalent to the introduction of non-linear patterns of dependence and independence between variables of Branching Quantifier Logic [7] or Independence Friendly Logic [9, 8, 14]: for example, both the Branching Quantifier Logic sentence

$$\begin{pmatrix} \forall x & \exists y \\ \forall z & \exists w \end{pmatrix} R(x, y, z, w)$$

and the Independence Friendly Logic sentence

$$\forall x \exists y \forall z (\exists w / x, y) R(x, y, z, w)$$

correspond in Dependence Logic to

$$\forall x \exists y \forall z \exists w (=\!(z,w) \wedge R(x,y,z,w)),$$

in the sense that all of these expressions are equivalent to the Skolem formula

$$\exists f \exists g \forall x \forall z R(x, f(x), z, g(z)).$$

As this example illustrates, the main peculiarity of Dependence Logic compared to the others above-mentioned logics lies in the fact that, in Dependence Logic, the notion of *dependence and independence between variables* is explicitly separated from the notion of quantification. This makes it an eminently suitable formalism for the formal analysis of the properties of *dependence itself* in a first-order setting, and some recent papers ([5, 2, 4]) explore the effects of replace dependence atoms with other similar primitives such as *independence atoms* [5], *multivalued dependence atoms* [2], or *inclusion* or *exclusion* atoms [3, 4].

The present work is an attempt to analyze the nature of the relationship between Dependence Logic and *Game Logics* such as the ones discussed in [15] and [19]. As we will see, there exists a profound connection between these two areas of logical research, which we will exploit in order to develop two game logics-inspired variants of Dependence Logic in which the interpretation, already implicit in the Game Theoretic Semantics of Dependence Logic, of formulas as assertions about *abilities* of an agent is brought to the forefront.

We will now recall some basic definitions and results which will be of some use for the rest of this work:

Definition 1.1 (Team) *Let M be a first-order model. A* team *over it is a set X of assignments, all with the same domain of variables $\texttt{Dom}(X) \subseteq \texttt{Var}$.*

Definition 1.2 (Relations corresponding to teams) *Let X be a team over M, and let $\vec{v}$ be a finite tuple of variables in its domain. Then $X(\vec{v})$ is the relation $\{s(\vec{v}) : s \in X\}$. Furthermore, we write $\texttt{Rel}(X)$ for $X(\texttt{Dom}(X))$.*

As is often the case for Dependence Logic, we will assume that all our formulas are in Negation Normal Form:

Definition 1.3 (Dependence Logic, Syntax) *Let Σ be a first-order signature. Then the set of all dependence logic formula with signature Σ is given by*

$$\phi ::= R\vec{t} \mid \neg R\vec{t} \mid \;=\!(t_1 \dots t_n) \mid \phi \vee \phi \mid \phi \wedge \phi \mid \exists v \phi \mid \forall v \phi$$

where R ranges over all relation symbols, $\vec{t}$ ranges over all tuples of terms of the appropriate arities, $t_1 \dots t_n$ range over all terms and v ranges over all variables in $\texttt{Var}$.

The set $\texttt{Free}(\phi)$ of all *free variables* of a formula ϕ is defined precisely as in First Order Logic, with the additional condition that all variables occurring in a dependence atom are free with respect to it.

Definition 1.4 (Dependence Logic, Semantics) *Let M be a first-order model, let X be a team over it, and let ϕ be a Dependence Logic formula with the same signature of M and with free variables in $\mathtt{Dom}(X)$. Then we say that X* satisfies *ϕ in M, and we write $M \models_X \phi$, if and only if*

TS-lit: *ϕ is a first-order literal and $M \models_s \phi$ for all $s \in X$;*

TS-dep: *ϕ is a dependence atom $=\!(t_1 \dots t_n)$ and any two assignments $s, s' \in X$ which assign the same values to $t_1 \dots t_{n-1}$ also assign the same value to t_n;*

TS-$\vee$: *ϕ is of the form $\psi_1 \vee \psi_2$ and there exist two teams Y_1 and Y_2 such that $X = Y_1 \cup Y_2$, $M \models_{Y_1} \psi_1$ and $M \models_{Y_2} \psi_2$;*

TS-$\wedge$: *ϕ is of the form $\psi_1 \wedge \psi_2$, $M \models_X \psi_1$ and $M \models_X \psi_2$;*

TS-$\exists$: *ϕ is of the form $\exists v \psi$ and there exists a function $F : X \to \mathtt{Dom}(M)$ such that $M \models_{X[F/v]} \psi$, where*

$$X[F/v] = \{s[F(s)/v] : s \in X\};$$

TS-$\forall$: *ϕ is of the form $\forall v \psi$ and $M \models_{X[M/v]} \psi$, where*

$$X[M/v] = \{s[m/v] : s \in X, m \in \mathtt{Dom}(M)\}.$$

The disjunction of Dependence Logic does not behave like the classical disjunction: for example, it is easy to see that $=\!(x) \vee =\!(x)$ is not equivalent to $=\!(x)$, as the former holds for the team $X = \{(x := 0), (x := 1)\}$ and the latter does not. However, it is possible to define the classical disjunction in terms of the other connectives:

Definition 1.5 (Classical Disjunction) *Let ψ_1 and ψ_2 be two Dependence Logic formulas, and let u_1 and u_2 be two variables not occurring in them. Then we write $\psi_1 \sqcup \psi_2$ as a shorthand for*

$$\exists u_1 \exists u_2 (=\!(u_1) \wedge =\!(u_2) \wedge ((u_1 = u_2 \wedge \psi_1) \vee (u_1 \neq u_2 \wedge \psi_2))).$$

Proposition 1.1 *For all formulas ψ_1 and ψ_2, all models M with at least two elements*[1] *whose signature contains that of ψ_1 and ψ_2 and all teams X whose domain contains the free variables of ψ_1 and ψ_2*

$$M \models_X \psi_1 \sqcup \psi_2 \Leftrightarrow M \models_X \psi_1 \text{ or } M \models_X \psi_2.$$

The following four proportions are from [16]:

Proposition 1.2 *For all models M and Dependence Logic formulas ϕ, $M \models_\emptyset \phi$.*

[1] In general, we will assume through this whole work that all first-order models which we are considering have at least two elements. As one-element models are trivial, this is not a very onerous restriction.

Proposition 1.3 (Downwards Closure) *If $M \models_X \phi$ and $Y \subseteq X$ then $M \models_Y \psi$.*

Proposition 1.4 (Locality) *If $M \models_X \phi$ and $X(\texttt{Free}(\phi)) = Y(\texttt{Free}(\phi))$ then $M \models_Y \phi$.*

Proposition 1.5 (From Dependence Logic to Σ_1^1) *Let $\phi(\vec{v})$ be a Dependence Logic formula with free variables in $\vec{v}$. Then there exists a Σ_1^1 sentence $\Phi(R)$ such that*

$$M \models_X \phi \Leftrightarrow M \models \Phi(X(\vec{v}))$$

for all suitable models M and for all nonempty teams X. Furthermore, in $\Phi(R)$ the symbol R occurs only negatively.

As proved in [13], there is also a converse for the last proposition:

Theorem 1.6 (From Σ_1^1 to Dependence Logic) *Let $\Phi(R)$ be a Σ_1^1 sentence in which R occurs only negatively. Then there exists a Dependence Logic formula $\phi(\vec{v})$, where $|\vec{v}|$ is the arity of R, such that*

$$M \models_X \phi \Leftrightarrow M \models \Phi(X(\vec{v}))$$

for all suitable models M and for all nonempty teams X whose domain contains $\vec{v}$.

Corollary 1.7 *Let P be any predicate symbol and let $\phi(\vec{v}, P)$ be any Dependence Logic formula with $\texttt{Free}(\phi) = \vec{v}$. Then there exists a Dependence Logic formula $\phi'(\vec{v})$ such that*

$$M \models_X \phi'(\vec{v}) \Leftrightarrow \exists P \text{ s.t. } M \models_X \phi(\vec{v}, P)$$

for all suitable models M and for all teams X whose domain contains $\vec{v}$.

Proof:
By Proposition 1.5, there exists a Σ_1^1 sentence $\Phi(R, P)$, in which R occurs only negatively, such that

$$M \models_X \phi(\vec{v}, P) \Leftrightarrow M \models \Phi(X(\vec{v}), P)$$

for all M and all nonempty X.

Now consider $\Phi'(R) := \exists P \Phi(R, P)$: by Theorem 1.6, there exists a formula $\phi'(\vec{v})$ such that, for all M and all nonempty X, $M \models_X \phi'(\vec{v})$ if and only if $M \models \Phi'(X(\vec{v}))$.

If X is instead empty then by Proposition 1.2 we have that $M \models_X \phi(\vec{v}, P)$ and $M \models_X \phi'(\vec{v})$; therefore, $\phi'(\vec{v})$ is the formula which we were looking for.
□

Definition 1.8 *Let $\phi(\vec{v}, P)$ be any Dependence Logic formula. Then we write $\exists P \phi(\vec{v}, P)$ for the Dependence Logic formula, whose existence follows from Corollary 1.7, such that*

$$M \models_X \exists P \phi(\vec{v}, P) \Leftrightarrow \exists P \text{ s.t. } M \models_X \phi(\vec{v}, P)$$

for all suitable models M and all (empty or nonempty) teams X.

1.2 Dynamic Game Logic

Game logics are logical formalisms which contain two different kinds of expressions:

1. *Game terms*, which are descriptions of games in terms of compositions of *atomic games*;

2. *Formulas*, which, in general, correspond to assertions about the abilities of players in games.

In this subsection, we are going to summarize the definition of a variant of Dynamic Game Logic [15].[2] Then, in the next subsection, we will discuss a remarkable connection between First-Order Logic and Dynamic Game Logic discovered by Johan van Benthem in [18].

One of the fundamental semantic concepts of Dynamic Game Logic is the notion of *forcing relation:*

Definition 1.9 (Forcing Relation) *Let S be a nonempty set of* states. *A* forcing relation *over S is a set $\rho \subseteq S \times \texttt{Parts}(S)$.*

In brief, a forcing relation specifies the abilities of a player in a perfect-information game: $(s, X) \in \rho$ if and only if the player has a strategy that guarantees that, whenever the initial position of the game is s, the terminal position of the game will be in X.

A (two-player) *game* is then defined as a pair of forcing relations satisfying some axioms:

Definition 1.10 (Game) *Let S be a nonempty set of states. A* game *over S is a pair (ρ^E, ρ^A) of forcing relations over S satisfying the following conditions for all $i \in \{E, A\}$, all $s \in S$ and all $X, Y \subseteq S$:*

Monotonicity: *If $(s, X) \in \rho^i$ and $X \subseteq Y$ then $(s, Y) \in \rho^i$;*

Consistency: *If $(s, X) \in \rho^E$ and $(s, Y) \in \rho^A$ then $X \cap Y \neq \emptyset$;*

Non-triviality: *$(s, \emptyset) \notin \rho^i$.*

Determinacy: *If $(s, X) \notin \rho^i$ then $(s, S \backslash X) \in \rho^j$, where $j \in \{E, A\} \backslash \{i\}$.*

Definition 1.11 (Game Model) *Let S be a nonempty set of states, let Φ be a nonempty set of* atomic propositions *and let Γ be a nonempty set of* atomic game symbols. *Then a* game model *over S, Φ and Γ is a triple $(S, \{(\rho_g^E, \rho_g^A) : g \in \Gamma\}, V)$, where (ρ_g^E, ρ_g^A) is a game over S for all $g \in \Gamma$ and where V is a valutation function associating each $p \in \Phi$ to a subset $V(p) \subseteq S$.*

[2]The main difference between this version and the one of Parikh's original paper lies in the absence of the iteration operator γ^* from our formalism. In this, we follow [18, 19].

The language of Dynamic Game Logic, as we already mentioned, consists of *game terms*, built up from atomic games, and of *formulas*, built up from atomic proposition. The connection between these two parts of the language is given by the *test* operation $\phi?$, which turns any formula ϕ into a test game, and the *diamond* operation, which combines a game term γ and a formula ϕ into a new formula $\langle\gamma, i\rangle\phi$ which asserts that agent i can guarantee that the game γ will end in a state satisfying ϕ.

Definition 1.12 (Dynamic Game Logic - Syntax) *Let Φ be a nonempty set of* atomic propositions *and let Γ be a nonempty set of* atomic game formulas*. Then the sets of all game terms γ and formulas ϕ are defined as*

$$\begin{array}{rcl} \gamma & ::= & g \mid \phi? \mid \gamma;\gamma \mid \gamma\cup\gamma \mid \gamma^d \\ \phi & ::= & \bot \mid p \mid \neg\phi \mid \phi\vee\phi \mid \langle\gamma, i\rangle\phi \end{array}$$

for p ranging over Φ, g ranging over Γ, and i ranging over $\{E, A\}$.

We already mentioned the intended interpretations of the test connective $\phi?$ and of the diamond connective $\langle\gamma, i\rangle\phi$. The interpretations of the other game connectives should be clear: γ^d is obtained by swapping the roles of the players in γ, $\gamma_1\cup\gamma_2$ is a game in which the existential player E chooses whether to play γ_1 or γ_2, and $\gamma_1;\gamma_2$ is the *concatenation* of the two games corresponding to γ_1 and γ_2 respectively.

Definition 1.13 (Dynamic Game Logic - Semantics) *Let $G = (S, \{(\rho^E_g, \rho^A_g) : g \in \Gamma, V)$ be a game model over S, Γ and Φ. Then for all game terms γ and all formulas ϕ of Dynamic Game Logic over Γ and Φ we define a game $\|\gamma\|_G$ and a set $\|\phi\|_G \subseteq S$ as follows:*

DGL-atomic-game: *For all $g \in G$, $\|g\|_G = (\rho^E_g, \rho^A_g)$;*

DGL-test: *For all formulas ϕ, $\|\phi?\|_G = (\rho^E, \rho^A)$, where*

- *$s\rho^E X$ iff $s \in \|\phi\|_G$ and $s \in X$;*

- *$s\rho^A X$ iff $s \notin \|\phi\|_G$ or $s \in X$*

for all $s \in S$ and all X with $\emptyset \neq X \subseteq S$;

DGL-concat: *For all game terms γ_1 and γ_2, $\|\gamma_1;\gamma_2\|_G = (\rho^E, \rho^A)$, where, for all $i \in \{E, A\}$ and for $\|\gamma_1\|_G = (\rho^E_1, \rho^A_1)$, $\|\gamma_2\|_G = (\rho^E_2, \rho^A_2)$,*

- *$s\rho^i X$ if and only if there exists a Z such that $s\rho^i_1 Z$ and for each $z \in Z$ there exists a set X_z satisfying $z\rho^i_2 X_1$ such that*

$$X = \bigcup_{z\in Z} X_z;$$

DGL-$\cup$: *For all game terms γ_1 and γ_2, $\|\gamma_1\cup\gamma_2\|_G = (\rho^E, \rho^A)$, where*

- $s\rho^E X$ *if and only if* $s\rho_1^E X$ *or* $s\rho_2^E X$*, and*
- $s\rho^A X$ *if and only if* $s\rho_1^A X$ *and* $s\rho_2^A X$

where, as before, $\|\gamma_1\|_G = (\rho_1^E, \rho_1^A)$ *and* $\|\gamma_2\|_G = (\rho_2^E, \rho_2^A)$*;*[3]

DGL-dual: *If* $\|\gamma\|_G = (\rho^E, \rho^A)$ *then* $\|\gamma^d\|_G = (\rho^A, \rho^E)$*;*

DGL-$\perp$: $\|\perp\|_G = \emptyset$*;*

DGL-atomic-pr: $\|p\|_G = V(p)$*;*

DGL-$\neg$: $\|\neg\phi\|_G = S \backslash \|\phi\|_G$*;*

DGL-$\vee$: $\|\phi_1 \vee \phi_2\|_G = \|\phi_1\|_G \cup \|\phi_2\|_G$*;*

DGL-$\diamond$: *If* $\|\gamma\|_G = (\rho^E, \rho^A)$ *then for all* ϕ*,*

$$\|\langle\gamma, i\rangle\phi\|_G = \{s \in S : \exists X_s \subseteq \|\phi\|_G \text{ s.t. } s\rho^i X_s\}.$$

If $s \in \|\phi\|_G$*, we say that* ϕ *is* satisfied *by* s *in* G *and we write* $M \models_s \phi$*.*

We will not discuss here the properties of this logic, or the vast amount of variants and extensions of it which have been developed and studied. It is worth pointing out, however, that [19] introduced a *Concurrent Dynamic Game Logic* that can be considered one of the main sources of inspiration for the Transition Logic that we will develop in Subsection 2.3.

1.3 The Representation Theorem

In this subsection, we will briefly recall a remarkable result from [18] which establishes a connection between Dynamic Game Logic and First-Order Logic.

In brief, as the following two theorems demonstrate, either of these logics can be seen as a special case of the other, in the sense that models and formulas of the one can be uniformly translated into models the other in a way which preserves satisfiability and truth:

Theorem 1.14 *Let* $G = (S, \{(\rho_g^E, \rho_g^A) : g \in \Gamma\}, V)$ *be any game model, let* ϕ *be any game formula for the same language, and let* $s \in S$*. Then it is possible to uniformly construct a first-order model* G^{FO}*, a first-order formula* ϕ^{FO} *and an assignment* s^{FO} *of* G^{FO} *such that*

$$G \models_s \phi \Leftrightarrow G^{FO} \models_{s^{FO}} \phi^{FO}.$$

[3] [19] gives the following alternative condition for the powers of the universal player:

- $s\rho^A X$ if and only if $X = Z_1 \cup Z_2$ for two Z_1 and Z_2 such that $s\rho_1^A Z_1$ and $s\rho_2^A Z_2$.

It is trivial to see that, if our games satisfy the monotonicity condition, this rules is equivalent to the one we presented.

Theorem 1.15 *Let M be any first order model, let ϕ be any first-order formula for the signature of M, and let s be an assignment of M. Then it is possible to uniformly construct a game model G^{DGL}, a game formula ϕ^{DGL} and a state s^{DGL} such that*

$$M \models_s \phi \Leftrightarrow G^{DGL} \models_{s^{DGL}} \phi^{DGL}.$$

We will not discuss here the proofs of these two results. Their *significance*, however, is something about which is necessary to spend a few words. In brief, what this back-and-forth representation between First Order Logic and Dynamic Game Logic tells us is that it is possible to understand First Order Logic as a *logic for reasoning about determined games*!

In the next sections, we will attempt to develop a similar result for the case of Dependence Logic.

2 Transition Logic

2.1 A Logic for Imperfect Information Games Against Nature

We will now define a variant of Dynamic Game Logic, which we will call *Transition Logic*. It deviates from the basic framework of Dynamic Game Logic in two fundamental ways:

1. It considers *one-player* games against Nature, instead of *two-player games* as is usual in Dynamic Game Logic;
2. It allows for *uncertainty* about the initial position of the game.

Hence, Transition Logic can be seen as a *decision-theoretic logic*, rather than a *game-theoretic* one: Transition Logic formulas, as we will see, correspond as assertions about the abilities of a single agent acting under uncertainty, instead of assertions about the abilities of agents interacting which each other.

In principle, it certainly possible to generalize the approach discussed here to multiple agents acting in situations of imperfect information, and doing so might cause interesting phenomena to surface; but for the time being, we will content ourselves with developing this formalism and proving the analogue of van Benthem's Representation Theorem for it.

Our first definition is a fairly straightforward generalization of the concept of forcing relation:

Definition 2.1 (Transition system) *Let S be a nonempty set of* states. *A* transition system *over S is a nonempty relation $\theta \subseteq \texttt{Parts}(S) \times \texttt{Parts}(S)$ satisfying the following requirements:*

Downwards Closure: *If $(X, Y) \in \theta$ and $X' \subseteq X$ then $(X', Y) \in \theta$;*

Monotonicity: *If $(X, Y) \in \theta$ and $Y \subseteq Y'$ then $(X, Y') \in \theta$;*

Non-creation: $(\emptyset, Y) \in \theta$ *for all* $Y \subseteq S$*;*

Non-triviality: *If* $X \neq \emptyset$ *then* $(X, \emptyset) \notin \theta$*.*

Informally speaking, a transition system specifies the abilities of an agent: for all $X, Y \subseteq S$ such that $(X, Y) \in \theta$, the agent has a strategy which guarantees that the output of the transition will be in Y whenever the input of the transition is in X.

The four axioms which we gave capture precisely this intended meaning, as we will see:

Definition 2.2 (Decision Game) *A* decision game *is a triple* $\Gamma = (S, E, O)$*, where* S *is a nonempty set of* states*,* E *is a nonempty set of* strategies *and* O *is an* outcome function *from* $S \times E$ *to* $\mathtt{Parts}(S)$*.*

If $s' \in O(s, e)$*, we say that* s' *is a* possible outcome *of* s *under* e*; if* $O(s, e) = \emptyset$*, we say that* e fails *on input* s*.*

Definition 2.3 (Abilities in a decision game) *Let* $\Gamma = (S, E, O)$ *be a decision game, and let* $X, Y \subseteq S$*. Then we say that* Γ allows *the transition* $X \to Y$*, and we write* $\Gamma : X \to Y$*, if and only if there exists a* $e \in E$ *such that* $\emptyset \neq O(s, e) \subseteq Y$ *for all* $s \in X$*.*[4]

Theorem 2.4 (Transition Systems and Abilities) *A set* $\theta \subseteq \mathtt{Parts}(S) \times \mathtt{Parts}(S)$ *is a transition system if and only if there exists a decision game* $\Gamma = (S, E, O)$ *such that*

$$(X, Y) \in \theta \Leftrightarrow \Gamma : X \to Y.$$

Proof:
Let $\theta = \{(X_i, Y_i) : i \in I)\}$ be a transition system, and let $\Gamma = (S, I, O)$ for

$$O(s, i) = \begin{cases} Y_i & \text{if } s \in X_i; \\ \emptyset & \text{otherwise.} \end{cases}$$

Suppose that $(X, Y) \in \theta$. If $X = \emptyset$, then $\Gamma : X \to Y$ follows at once by definition. If instead $X \neq \emptyset$, by **non-triviality** we have that Y is nonempty too, and furthermore $(X, Y) = (X_i, Y_i)$ for some $i \in I$. Then $O(s, i) = Y_i \neq \emptyset$ for all $s \in X_i$, as required.

Now suppose that $\Gamma : X \to Y$. Then there exists a $i \in I$ such that $\emptyset \neq O(s, i) \subseteq Y$ for all $s \in X$. If $X \neq \emptyset$, this implies that $X \subseteq X_i$ and $Y_i \subseteq Y$. Hence, by **monotonicity** and **downwards closure**, $(X, Y) \in \theta$, as required. If instead $X = \emptyset$, then by **non-creation** we have again that $(X, Y) \in \theta$.

Conversely, consider a decision game $\Gamma = (S, E, O)$. Then the set of its abilities satisfies our four axioms:

Downwards Closure: Suppose that $\Gamma : X \to Y$ and that $X' \subseteq X$. By definition, there exists a $e \in E$ such that $\emptyset \neq O(s, e) \subseteq Y$ for all $s \in X$. But then the same holds for all $s \in X'$, and hence $\Gamma : X' \to Y$.

[4] That is, if and only if our agent has a strategy which guarantees that the outcome will be in Y whenever the input is in X.

Monotonicity: Suppose that $\Gamma : X \to Y$ and that $Y \subseteq Y'$. By definition, there exists a $e \in E$ such that $\emptyset \neq O(s,e) \subseteq Y$ for all $s \in X$. But then, for all such s, $O(s,e) \subseteq Y'$ too, and hence $\Gamma : X \to Y'$.

Non-creation: Let $Y \subseteq S$ and let $e \in E$ be any strategy. Then trivially $\emptyset \neq O(s,e) \subseteq Y$ for all $s \in \emptyset$, and hence $\Gamma : \emptyset \to Y$.

Non-triviality: Let $s_0 \in X$, and suppose that $\Gamma : X \to Y$. Then there exists a e such that $\emptyset \neq O(s,e) \subseteq Y$ for all $s \in X$, and hence in particular $\emptyset \neq O(s_0,e) \subseteq Y$. Therefore, Y is nonempty.

□

Definition 2.5 (Trump) *Let S be a nonempty set of states. A* trump *over S is a nonempty, downwards closed family of subsets of S.*

Whereas a transition system describes the abilities of an agent to transition from a set of possible initial states to a set of possible terminal states, a trump describes the agent's abilities to reach *some* terminal state from a set of possible initial states:[5]

Proposition 2.1 *Let θ be a transition system and let $Y \subseteq S \neq \emptyset$. Then $\texttt{reach}(\theta, Y) = \{X \mid (X,Y) \in \theta\}$ forms a trump. Conversely, for any trump $\mathcal{X}$ over S there exists a transition system θ such that $\mathcal{X} = \texttt{reach}(\theta, Y)$ for any nonempty $Y \subseteq S$.*

Proof:
Let θ be a transition system. Then if $(X,Y) \in \theta$ and $X' \subseteq X$, by downwards closure we have at once that $(X',Y) \in \theta$. Furthermore, $(\emptyset, Y) \in \theta$ for any Y. Hence, $\texttt{reach}(\theta, Y)$ is a trump, as required.

Conversely, let $\mathcal{X} = \{X_i : i \in I\}$ be a trump. Then define θ as

$$\theta = \{(A,B) : \emptyset \neq B \subseteq S, \exists i \in I \text{ s.t. } A \subseteq X_i\} \cup \{(\emptyset, \emptyset)\}$$

It is easy to see that θ is a transition system; and by construction, for $Y \neq \emptyset$ we have that $(A,Y) \in \theta \Leftrightarrow \exists i$ s.t. $A \subseteq X_i \Leftrightarrow A \in \mathcal{X}$, where we used the fact that $\mathcal{X}$ is downwards closed.
□

We can now define the syntax and semantics of Transition Logic:

Definition 2.6 (Transition Model) *Let Φ be a set of* atomic propositional symbols *and let Θ be a set of* atomic transition symbols*. Then a* transition model *is a tuple $T = (S, \{\theta_t : t \in \Theta\}, V)$, where S is a nonempty set of states, θ_t is a transition system over S for any $t \in \Theta$, and V is a function sending each $p \in \Phi$ into a trump of S.*

[5] The term "trump" is taken from [11], where it is used to describe the set of all teams which satisfy a given formula.

Definition 2.7 (Transition Logic - Syntax) *Let Φ be a set of atomic propositions and let Θ be a set of atomic transitions. Then the* transition terms *and* formulas *of our language are defined respectively as*

$$\begin{array}{rcl} \tau & ::= & t \mid \phi? \mid \tau \otimes \tau \mid \tau \cap \tau \mid \tau ; \tau \\ \phi & ::= & \top \mid p \mid \phi \vee \phi \mid \phi \wedge \phi \mid \langle \tau \rangle \phi \end{array}$$

where t ranges over Θ and p ranges over Φ.

Definition 2.8 (Transition Logic - Semantics) *Let $T = (S, \{\theta_t : t \in \Theta\}, V)$ be a transition model, let τ be a transition term, and let $X, Y \subseteq S$. Then we say that τ* allows *the transition from X to Y, and we write $T \models_{X \to Y} \tau$, if and only if*

TL-atomic-tr: *$\tau = t$ for some $t \in \Theta$ and $(X, Y) \in \theta_t$;*

TL-test: *$\tau = \phi?$ for some transition formula ϕ such that $T \models_X \phi$, and $X \subseteq Y$;*

TL-$\otimes$: *$\tau = \tau_1 \otimes \tau_2$, and $X = X_1 \cup X_2$ for two X_1 and X_2 such that $T \models_{X_1 \to Y} \tau_1$ and $T \models_{X_2 \to Y} \tau_2$;*

TL-$\cap$: *$\tau = \tau_1 \cap \tau_2$, $T \models_{X \to Y} \tau_1$ and $T \models_{X \to Y} \tau_2$;*

TL-concat: *$\tau = \tau_1 ; \tau_2$ and there exists a $Z \subseteq S$ such that $T \models_{X \to Z} \tau_1$ and $T \models_{Z \to Y} \tau_2$.*

Analogously, let ϕ be a transition formula, and let $X \subseteq S$. Then we say that X satisfies *ϕ, and we write $T \models_X \phi$, if and only if*

TL-$\top$: *$\phi = \top$;*

TL-atomic-pr: *$\phi = p$ for some $p \in \Phi$ and $X \in V(p)$;*

TL-$\vee$: *$\phi = \psi_1 \vee \psi_2$ and $T \models_X \psi_1$ or $T \models_X \psi_2$;*

TL-$\wedge$: *$\phi = \psi_1 \wedge \psi_2$, $T \models_X \psi_1$ and $T \models_X \psi_2$;*

TL-$\diamond$: *$\phi = \langle \tau \rangle \psi$ and there exists a Y such that $T \models_{X \to Y} \tau$ and $T \models_Y \psi$.*

Proposition 2.2 *For any transition model T, transition term τ and transition formula ϕ, the set*

$$\|\tau\|_T = \{(X, Y) : T \models_{X \to Y} \tau\}$$

is a transition system and the set

$$\|\phi\|_T = \{X : T \models_X \phi\}$$

is a trump.

Proof:
By induction.
□
We end this subsection with a few simple observations about this logic.

First of all, we did not take the negation as one of the primitive connectives. Indeed, Transition Logic, much like Dependence Logic, has an intrinsically *existential* character: it can be used to reason about which sets of possible states an agent *may* reach, but not to reason about which ones such an agent *must* reach. There is of course no reason, in principle, why a negation could not be added to the language, just as there is no reason why a negation cannot be added to Dependence Logic, thus obtaining the far more powerful *Team Logic* [17, 12]: however, this possible extension will not be studied in this work.

The connectives of Transition Logic are, for the most part, very similar to those of Dynamic Game Logic, and their interpretation should pose no difficulties. The exception is the *tensor operator* $\tau_1 \otimes \tau_2$, which substitutes the game union operator $\gamma_1 \cup \gamma_2$ and which, while sharing roughly the same informal meaning, behaves in a very different way from the semantic point of view (for example, it is not in general idempotent!)

The decision game corresponding to $\tau_1 \otimes \tau_2$ can be described as follows: first the agent chooses an index $i \in \{1, 2\}$, then he or she picks a strategy for τ_i and plays accordingly. However, the choice of i may be a function of the initial state: hence, the agent can guarantee that the output state will be in Y whenever the input state is in X only if he or she can split X into two subsets X_1 and X_2 and guarantee that the state in Y will be reached from any state in X_1 when τ_1 is played, and from any state in X_2 when τ_2 is played.

It is also of course possible to introduce a "true" choice operator $\tau_1 \cup \tau_2$, with semantical condition

TL-$\cup$: $T \models_{X \to Y} \tau_1 \cup \tau_2$ iff $T \models_{X \to Y} \tau_1$ or $T \models_{X \to Y} \tau_2$;

but we will not explore this possibility any further in this work, nor we will consider any other possible connectives such as, for example, the iteration operator

TL-$*$: $T \models_{X \to Y} \tau^*$ iff there exist $n \in \mathbb{N}$ and $Z_0 \ldots Z_n$ such that $Z_0 = X$, $Z_n = Y$ and $T \models_{Z_i \to Z_{i+1}} \tau$ for all $i \in 1 \ldots n-1$.

2.2 A Representation Theorem for Dependence Logic

This subsection contains the central result of this work, that is, an analogue of van Benthem's result (Theorems 1.14 and 1.15 here) for Dependence Logic and Transition Logic.

Representing Dependence Logic models and formulas in Transition Logic is fairly simple:

Definition 2.9 (M^{TL}) *Let M be a first-order model. Then $M^{TL} = (S, \{\theta_{\exists v}, \theta_{\forall v} : v \in \mathtt{Var}\}, V)$ is the transition model such that*

- *S is the set of all teams over M;*
- *For any variable v, $\theta_{\exists v} = \{(X,Y) : \exists F$ s.t. $X[F/v] \subseteq Y\}$ and $\theta_{\forall v} = \{(X,Y) : X[M/v] \subseteq Y\}$;*
- *For any first-order literal or dependence atom α, $V(\alpha) = \{X : M \models_X \phi\}$.*

Definition 2.10 (ϕ^{TL}) *Let ϕ be a Dependence Logic formula. Then ϕ^{TL} is the transition term defined as follows:*

1. *If ϕ is a literal or a dependence atom, $\phi^{TL} = \phi?$;*
2. *If $\phi = \psi_1 \vee \psi_2$, $\phi^{TL} = (\psi_1)^{TL} \otimes (\psi_2)^{TL}$;*
3. *If $\phi = \psi_1 \wedge \psi_2$, $\phi^{TL} = (\psi_1)^{TL} \wedge (\psi_2)^{TL}$;*
4. *If $\phi = \exists v \psi$, $\phi^{TL} = \exists v; (\psi)^{TL}$;*
5. *If $\phi = \forall v \psi$, $\phi^{TL} = \forall v; (\psi)^{TL}$.*

Theorem 2.11 *For all first-order models M, teams X and formulas ϕ, the following are equivalent:*

- *$M \models_X \phi$;*
- *$\exists Y$ s.t. $M^{TL} \models_{X \to Y} \phi^{TL}$;*
- *$M^{TL} \models_X \langle \phi^{TL} \rangle \top$;*
- *$M^{TL} \models_{X \to S} \phi^{TL}$.*

Proof:
We show, by structural induction on ϕ, that the first condition is equivalent to the last one. The equivalences between the last one and the second and third ones are then trivial.

1. If ϕ is a literal or a dependence atom, $M^{TL} \models_{X \to S} \phi?$ if and only if $X \in V(\phi)$, that is, if and only if $M \models_X \phi$;
2. $M^{TL} \models_{X \to S} (\psi_1)^{TL} \otimes (\psi_2)^{TL}$ if and only if $X = X_1 \cup X_2$ for two $X_1, X_2 \subseteq S$ such that $M^{TL} \models_{X_1 \to S} (\psi_1)^{TL}$ and $M^{TL} \models_{X_2 \to S} (\psi_2)^{TL}$. By induction hypothesis, this can be the case if and only if $M \models_{X_1} \psi_1$ and $M \models_{X_2} \psi_2$, that is, if and only if $M \models_X \psi_1 \vee \psi_2$.
3. $M^{TL} \models_{X \to S} (\psi_1)^{TL} \wedge (\psi_2)^{TL}$ if and only if $M^{TL} \models_{X \to S} (\psi_1)^{TL}$ and $M^{TL} \models_{X \to S} (\psi_2)^{TL}$, that is, by induction hypothesis, if and only if $M \models_X \psi_1 \wedge \psi_2$.
4. $M^{TL} \models_{X \to S} \exists v; (\psi)^{TL}$ if and only if there exists a Y such that $Y \supseteq X[F/v]$ for some F and $M^{TL} \models_{Y \to S} \psi$. By induction hypothesis and downwards closure, this can be the case if and only if $M \models_{X[F/v]} \psi$ for some F, that is, if and only if $M \models_X \exists v \psi$;

5. $M^{TL} \models_{X \to S} \forall v; (\psi)^{TL}$ if and only if $M^{TL} \models_{Y \to S} (\psi)^{TL}$ for some $Y \supseteq X[M/v]$, that is, if and only if $M \models_{X[M/v]} \psi$, that is, if and only if $M \models_X \forall v \psi$.

$\square$

Representing Transition Models, game terms and formulas in Dependence Logic is somewhat more complex:

Definition 2.12 (T^{DL}) *Let $T = (S, (\theta_t : t \in \Theta), V)$ be a transition model. Furthermore, for any $t \in \Theta$, let $\theta_t = \{(X_i, Y_i) : i \in I_t\}$, and, for any $p \in \Phi$, let $V(p) = \{X_j : j \in J_p\}$. Then T^{DL} is the first-order model with domain*[6] *$S \uplus \biguplus\{I_t : t \in \Theta\} \uplus \biguplus\{J_p : p \in \Phi\}$ whose signature contains*

- *For every $t \in \Theta$, a ternary relation R_t whose interpretation is $\{(i, x, y) : i \in I_t, x \in X_i, y \in Y_i\}$;*
- *For every $p \in \Phi$, a binary relation V_p whose interpretation is $\{(j, x) : j \in J_p, x \in X_j\}$.*

Definition 2.13 (ϕ_x^{DL} and τ_x^{DL}) *For any formula ϕ, transition term τ, variable x and unary relation symbol P the Dependence Logic formulas ϕ_x^{DL} and $\tau_x^{DL}(P)$ are defined as follows:*

1. *$\top_x^{DL}$ is $\top$;*
2. *For all $p \in \Phi$, p_x^{DL} is $\exists j(=(j) \wedge V_p(j, x))$;*
3. *$(\psi_1 \vee \psi_2)_x^{DL}$ is $(\psi_1)_x^{DL} \sqcup (\psi_2)^{DL}$, where $\sqcup$ is the classical disjunction introduced in Definition 1.5;*
4. *$(\psi_1 \wedge \psi_2)_x^{DL}$ is $(\psi_1)_x^{DL} \wedge (\psi_2)_x^{DL}$;*
5. *$(\langle\tau\rangle\psi)_x^{DL}$ is $\exists P((\tau)_x^{DL}(P) \wedge \forall y(\neg Py \vee (\psi)_y^{DL}))$, where the second-order existential quantifier is a shorthand for the construction described in Definition 1.8 and y is a new and unused variable;*

1. *For all $t \in \Theta$, $t_x^{DL}(P)$ is $\exists i(=(i) \wedge \exists y(R_t(i, x, y)) \wedge \forall y(\neg R_t(i, x, y) \vee Py))$;*
2. *For all formulas ϕ, $(\phi?)_x^{DL}(P)$ is $\phi_x^{DL} \wedge Px$;*
3. *$(\tau_1 \otimes \tau_2)_x^{DL}(P) = (\tau_1)_x^{DL}(P) \vee (\tau_2)_x^{DL}(P)$;*
4. *$(\tau_1 \cap \tau_2)_x^{DL}(P) = (\tau_1)_x^{DL}(P) \wedge (\tau_2)_x^{DL}(P)$;*
5. *$(\tau_1; \tau_2)_x^{DL}(P) = \exists Q((\tau_1)_x^{DL}(Q) \wedge \forall y(\neg Qy \vee (\tau_2)_y^{DL}(P)))$, where y a new and unused variable.*

[6] Here we write $A \uplus B$ for the *disjoint union* of the sets A and B.

Theorem 2.14 *For all transition models $T = (S, (\theta_t : t \in \Theta), V)$, transition terms τ, transition formulas ϕ, variables x, sets $P \subseteq S$ and teams X over T^{DL},*

$$T^{DL} \models_X \phi^{DL}_x \Leftrightarrow T \models_{X(x)} \phi$$

and

$$T^{DL} \models_X \tau^{DL}_x(P) \Leftrightarrow T \models_{X(x) \to P} \tau.$$

Proof:
The proof is by structural induction on terms and formulas.

Let us first consider the cases corresponding to formulas:

1. For all teams X, $T^{DL} \models_X \top$ and $T \models_{X(x)} \top$, as required;

2. Suppose that $T^{DL} \models_X \exists j(=(j) \wedge V_p(j, x))$. Then there exists a $m \in \mathtt{Dom}(T^{DL})$ such that $T^{DL} \models_{X[m/j]} V_p(j, x)$. Hence, we have that $X(x) \subseteq X_m \in V(p)$; and, by downwards closure, this implies that $X(x) \in V(p)$, and hence that $T \models_{X(x)} p$ as required.

   Conversely, suppose that $T \models_{X(x)} p$. Then $X(x) \in V(p)$, and hence $X(x) = X_m$ for some $m \in J_p$. Then we have by definition that $T^{DL} \models_{X[m/j]} V_p(j, x)$, and finally that $T^{DL} \models_X T_x(p)$.

3. By Proposition 1.1, $T^{DL} \models_X (\psi_1 \vee \psi_2)^{DL}_x$ if and only if $T^{DL} \models_X (\psi_1)^{DL}_x$ or $T^{DL} \models_X (\psi_2)^{DL}_x$. By induction hypothesis, this is the case if and only if $T \models_{X(x)} \psi_1$ or $T \models_{X(x)} \psi_2$, that is, if and only if $T \models_{X(x)} \psi_1 \vee \psi_2$.

4. $T^{DL} \models_X (\psi_1 \wedge \psi_2)^{DL}_x$ if and only if $T^{DL} \models_X (\psi_1)^{DL}_x$ and $T^{DL} \models_x (\psi_2)^{DL}_x$, that is, by induction hypothesis, if and only if $T \models_X \psi_1 \wedge \psi_2$.

5. $T^{DL} \models_X (\langle \tau \rangle \psi)^{DL}_x$ if and only if there exists a P such that $T^{DL} \models_X (\tau)^{DL}_x(P)$ and $T^{DL} \models_{X[T^{DL}/y]} \neg Py \vee (\psi)^{DL}_y$. By induction hypothesis, the first condition holds if and only if $T \models_{X(x) \to P} \tau$. As for the second one, it holds if and only if $X[T^{DL}/y] = Y_1 \cup Y_2$ for two Y_1, Y_2 such that $T^{DL} \models_{Y_1} \neg Py$ and $T^{DL} \models_{Y_2} \tau_y(\psi)$. But then we must have that $T \models_{Y_2(y)} \psi$ and that $P \subseteq Y_2(y)$; therefore, by downwards closure, $T \models_P \psi$ and finally $T \models_{X(x)} \langle \tau \rangle \psi$.

   Conversely, suppose that there exists a P such that $T \models_{X(x) \to P} \tau$ and $T \models_P \psi$; then by induction hypothesis we have that $T^{DL} \models_X (\tau)^{DL}_x(P)$ and that $T^{DL} \models_{X[T^{DL}/y]} \neg Py \vee (\psi)^{DL}_x$, and hence $T^{DL} \models_X (\langle \tau \rangle \psi)^{DL}_x$.

Now let us consider the cases corresponding to transition terms:

1. Suppose that $T^{DL} \models_X \exists i(=(i) \wedge \exists y(R_t(i, x, y)) \wedge \forall y(\neg R_t(i, x, y) \vee Py))$. If $X = \emptyset$ then $X(x) = \emptyset$, and hence by **non-creation** we have that $(X(x), P) = (\emptyset, P) \in \theta_t$, as required.

   Let us assume instead that $X \neq \emptyset$. Then, by hypothesis, there exists a $m \in \mathtt{Dom}(T^{DL})$ such that

- There exists a F such that $T^{DL} \models_{X[m/i][F/y]} R_t(i, x, y)$;
- $T^{DL} \models_{X[m/i][T^{DL}/y]} \neg R_t(i, x, y) \vee Py$.

From the first condition it follows that for every $p \in X(x)$ there exists a q such that $R_t(m, p, q)$: therefore, by the definition of R_t, every such p must be in X_m.

From the second condition it follows that whenever $R_t(m, p, q)$ and $p \in X(x) \subseteq X_m$, $q \in P$; and, since $X(x) \neq \emptyset$, this implies that $Y_m \subseteq P$ by the definition of R_t.

Hence, by **monotonicity** and **downwards closure**, we have that $(X(x), P) \in \theta_t$ and that $T \models_{X(x) \to P} t$, as required.

Conversely, suppose that $(X(x), P) = (X_m, Y_m) \in \theta_t$ for some $m \in I_t$. If $X(x) = \emptyset$ then $X = \emptyset$, and hence by Proposition 1.2 we have that $T^{DL} \models_X t^{DL}_x(P)$, as required. Otherwise, by **non-triviality**, $P = Y_m \neq \emptyset$. Let now $p \in P$ be any of its elements and let $F(s) = p$ for all $p \in X[m/i]$: then $M \models_{X[m/i][F/y]} R_t(i, x, y)$, as any assignment of this team sends x to some element of X_m and y to $p \in Y_m$. Furthermore, let $s \in X(x) = X_m$, and let q be such that $R_t(m, s(x), q)$: then $q \in Y_m = P$, and hence $M \models_{X[m/i][T^{DL}/y]} \neg R_t(i, x, y) \vee Py$. So, in conclusion, $M \models_X t^{DL}_x(P)$, as required.

2. $T^{DL} \models_X \phi^{DL}_x \wedge Px$ if and only if $T \models_{X(x)} \phi$ and $X(x) \subseteq P$, that is, if and only if $T \models_{X(x) \to P} \phi?$.

3. $T^{DL} \models_X (\tau_1)^{DL}_x(P) \vee (\tau_2)^{DL}_x(P)$ if and only if $X = X_1 \cup X_2$ for two X_1, X_2 such that

   - $X = X_1 \cup X_2$, and therefore $X(x) = X_1(x) \cup X_2(x)$;
   - $T^{DL} \models_{X_1} (\tau_1)^{DL}_x(P)$, that is, by induction hypothesis, $T \models_{X_1(x) \to P} \tau_1$;
   - $T^{DL} \models_{X_2} (\tau_2)^{DL}_x(P)$, that is, by induction hypothesis, $T \models_{X_2(x) \to P} \tau_2$;

   Hence, if $T^{DL} \models_X (\tau_1 \otimes \tau_2)^{DL}_x(P)$ then $T \models_{X(x) \to P} \tau_1 \otimes \tau_2$.

   Conversely, if $X(x) = A \cup B$ for two A, B such that $T \models_{A \to P} \tau_1$ and $T \models_{B \to P} \tau_2$, let

$$\begin{aligned} X_1 &= \{s \in X : s(x) \in A\} \\ X_2 &= \{s \in X : s(x) \in B\}. \end{aligned}$$

   Clearly $X = X_1 \cup X_2$, and furthermore by induction hypothesis $T^{DL} \models_{X_1} (\tau_1)^{DL}_x(P)$ and $T^{DL} \models_{X_2} (\tau_2)^{DL}_x(P)$. Hence, $T^{DL} \models_X (\tau_1 \otimes \tau_2)^{DL}_x(P)$, as required.

4. $T^{DL} \models_X (\tau_1 \cap \tau_2)^{DL}_x(P)$ if and only if $T^{DL} \models_X (\tau_1)^{DL}_x(P)$ and $T^{DL} \models_X (\tau_2)^{DL}_x(P)$, that is, by induction hypothesis, if and only if $T \models_{X(x) \to P} \tau_1 \cap \tau_2$.

5. $T^{DL} \models_X \exists Q((\tau_1)^{DL}_x(Q) \wedge \forall y(\neg Qy \vee (\tau_2)^{DL}_y(P)))$ if and only if there exists a Q such that $T \models_{X(x)\to Q} \tau_1$ and there exists a $Q' \supseteq Q$ such that $T \models_{Q' \to P}$ τ_2. By downwards closure, if this is the case then $T \models_{Q\to P} \tau_2$ too, and hence $T \models_{X(x)\to P} \tau_1; \tau_2$, as required.

   Conversely, suppose that there exists a Q such that $T \models_{X(x)\to Q} \tau_1$ and $T \models_{Q\to P} \tau_2$. Then, by induction hypothesis $T^{DL} \models_X (\tau_1)^{DL}_x(Q)$; and furthermore, $X[T^{DL}/y]$ can be split into

$$Z_1 = \{s \in X[T^{DL}/y] : s(y) \notin Q\}$$

   and

$$Z_2 = \{s \in X[T^{DL}/y] : s(y) \in Q\}$$

   It is trivial to see that $T^{DL} \models_{Z_1} \neg Qy$; and furthermore, since $Z_2(y) = Q$ and $T \models_{Q\to P} \tau_2$, by induction hypothesis we have that $T^{DL} \models_{Z_2}$ $(\tau_2)^{DL}_y$. Thus $T^{DL} \models_{X[T^{DL}/y]} \forall y(\neg Qy \vee (\tau_2)^{DL}_y(P))$ and finally $T^{DL} \models_X$ $(\tau_1; \tau_2)^{DL}_x(P)$, and this concludes the proof.

□

2.3 Transition Dependence Logic

Just as van Benthem's representation theorem, which we recalled in Subsection 1.3, allows one to reinterpret First Order Logic as a logic of perfect information two-player games, Theorems 2.11 and 2.14 of Subsection 2.2 permit us to understand Dependence Logic as a logic of imperfect-information decision problems.

However, the language of Dependence Logic, as described in Subsection 1.1, does very little to support this interpretation. We may certainly associate a Dependence Logic sentence such as, for example, $\forall x \exists y (=(y, f(x)) \wedge Pxy)$, to a certain game of imperfect information, and then establish a correspondence between the truth of the sentence and certain abilities of an agent in this game; but this interpretation - legitimate though it may be from a semantic perspective - does not appear to arise entirely naturally from the syntactical structure of the sentence.

But by exploiting of the representation of Dependence Logic inside of Transition Logic of Definitions 2.9 and 2.10, it is not difficult to define a variant of Dependence Logic in which this interpretation is manifested at the syntactical level itself:

Definition 2.15 (Transition Dependence Logic - Syntax) *Let Σ be a first-order signature. Then the sets of all* transition terms *and of all* formulas *of Dependence Transition Logic are given by the rules*

$$\begin{array}{lcl} \tau & ::= & \exists v \mid \forall v \mid \phi? \mid \tau \otimes \tau \mid \tau \cap \tau \mid \tau; \tau \\ \phi & ::= & R\vec{t} \mid \neg R\vec{t} \mid\ =(t_1 \ldots t_n) \mid \phi \vee \phi \mid \phi \wedge \phi \mid \langle \tau \rangle \phi. \end{array}$$

where v ranges over all variables in $\mathtt{Var}$, R ranges over all relation symbols of the signature, $\vec{t}$ ranges over all tuples of terms of the required arities, n ranges over $\mathbb{N}$ and $t_1 \ldots t_n$ range over the terms of our signature.

Definition 2.16 (Transition Dependence Logic - Semantics) *Let M be a first-order model, let τ be a first-order transition term of the same signature, and let X and Y be teams over M. Then we say that the transition $X \to Y$ is* allowed *by τ in M, and we write $M \models_{X \to Y} \tau$, if and only if*

TDL-$\exists$: *τ is of the form $\exists v$ for some $v \in \mathtt{Var}$ and there exists a F such that $X[F/v] \subseteq Y$;*

TDL-$\forall$: *τ is of the form $\forall v$ for some $v \in \mathtt{Var}$ and $X[M/v] \subseteq Y$;*

TDL-test: *τ is of the form $\phi?$, $M \models_X \phi$, and $X \subseteq Y$;*

TDL-$\otimes$: *τ is of the form $\tau_1 \otimes \tau_2$ and $X = X_1 \cup X_2$ for some X_1 and X_2 such that $M \models_{X_1 \to Y} \tau_1$ and $M \models_{X_2 \to Y} \tau_2$;*

TDL-$\cap$: *τ is of the form $\tau_1 \cap \tau_2$, $M \models_{X \to Y} \tau_1$ and $M \models_{X \to Y} \tau_2$;*

TDL-concat: *τ is of the form $\tau_1; \tau_2$ and there exists a team Z such that $M \models_{X \to Z} \tau_1$ and $M \models_{Z \to Y} \tau_2$.*

Similarly, if ϕ is a formula and X is a team with domain $\mathtt{Var}$. Then we say that X satisfies *ϕ in M, and we write $M \models_X \phi$, if and only if*

TDL-lit: *ϕ is a first-order literal and $M \models_s \phi$ in the usual first-order sense for all $s \in X$;*

TDL-dep: *ϕ is a dependence atom $=(t_1 \ldots t_n)$ and any two $s, s' \in X$ which assign the same values to $t_1 \ldots t_{n-1}$ also assign the same value to t_n;*

TDL-$\vee$: *ϕ is of the form $\phi_1 \vee \phi_2$ and $M \models_X \phi_1$ or $M \models_X \phi_2$;*

TDL-$\wedge$: *ϕ is of the form $\phi_1 \wedge \phi_2$, $M \models_X \phi_1$ and $M \models_X \phi_2$;*

TDL-$\diamond$: *ϕ is of the form $\langle \tau \rangle \psi$ and there exists a Y such that $M \models_{X \to Y} \tau$ and $M \models_Y \psi$.*

It is not difficult to see, on the basis of the results of the previous section, that this new variant of Dependence Logic is equivalent to the usual one:

Theorem 2.17 *For every Dependence Logic formula ϕ there exists a Transition Dependence Logic transition term τ_ϕ such that*

$$M \models_X \phi \Leftrightarrow \exists Y \text{ s.t. } M \models_{X \to Y} \tau_\phi \Leftrightarrow M \models_X \langle \tau_\phi \rangle \top$$

for all first-order models M and teams X.

Proof:
τ_ϕ is defined by structural induction on ϕ, as follows:

1. If ϕ is a first-order literal or a dependence atom then $\tau_\phi = \phi?$;
2. If ϕ is $\phi_1 \vee \phi_2$ then $\tau_\phi = \tau_{\phi_1} \otimes \tau_{\phi_2}$;
3. If ϕ is $\phi_1 \wedge \phi_2$ then $\tau_\phi = \tau_{\phi_1} \cap \tau_{\phi_2}$;
4. If ϕ is $\exists v \psi$ then $\tau_\phi = \exists v; \tau_\psi$;
5. If ϕ is $\forall v \psi$ then $\tau_\phi = \forall v; \tau_\psi$.

It is then trivial to verify, again by induction on ϕ, that $M \models_X \phi$ if and only if $M \models_X \langle \tau_\phi \rangle \top$, as required.
□

Theorem 2.18 *For every Transition Dependence Logic formula ϕ there exists a Dependence Logic formula ϕ' such that*

$$M \models_X \phi \Leftrightarrow M \models_X \phi'$$

for all first-order models M and teams X.

Proof:
(Sketch)
Translate ϕ into Σ^1_1, and then apply Theorem 1.6.
□
However, in a sense, Transition Dependence Logic allows one to consider subtler distinctions than Dependence Logic does. The formula $\forall x \exists y (=(y, f(x)) \wedge Pxy)$, for example, could be translated as any of

- $\langle \forall x; \exists y \rangle (=(y, f(x)) \wedge Pxy)$;
- $\langle \forall x; \exists y \rangle \langle =(y, f(x))? \rangle Pxy$;
- $\langle \forall x; \exists y \rangle \langle Pxy? \rangle =(y, f(x))$;
- $\langle \forall x; \exists y \rangle \langle (Pxy?) \cap (=(y, f(x))?) \rangle \top$.

The intended interpretations of these formulas are rather different, even though they happen to be satisfied by the same teams: and for this reason, Transition Dependence Logic may be thought of as a *refinement* of Dependence Logic proper, even though it has exactly the same expressive power.

3 Dynamic Semantics

3.1 Dynamic Predicate Logic

Dynamic Semantics is the name given to a family of semantical frameworks which subscribe to the following principle ([6]):

> *The meaning of a sentence does not lie in its truth conditions, but rather in the way it changes (the representation of) the information of the interpreter.*

In various forms, this intuition can be found prefigured in some of the later work of Ludwig Wittgenstein, as well as in the research of philosophers of language such as Austin, Grice, Searle, Strawson and others ([1]); but its formal development can be traced back to the work of Groenendijk and Stokhof about the proper treatment of pronouns in formal linguistics ([6]).

We refer to [1] for a comprehensive analysis of the linguistic issues which caused such a development, as well as for a description of the ways in which this framework was adapted in order to model presuppositions, questions/answers and other phenomena; here we will only present a formulation of *dynamic predicate semantics*, the alternative semantics for first-order logic which was developed in the above mentioned paper by Groenendijk and Stokhof.

Definition 3.1 (Dynamic Semantics for First-Order Logic) *Let ϕ be a first-order formula, let M be a suitable first-order model and let s and s' be two assignments. Then we say that the transition from s to s' is* allowed *by ϕ in M, and we write $M \models_{s \to s'} \phi$, if and only if*

DPL-atom: *ϕ is an atomic formula, $s = s'$ and $M \models_s \phi$ in the usual sense;*

DPL-$\neg$: *ϕ is of the form $\neg\psi$, $s = s'$ and for all assignments h, $M \not\models_{s \to h} \psi$;*

DPL-$\wedge$: *ϕ is of the form $\psi_1 \wedge \psi_2$ and there exists a h such that $M \models_{s \to h} \psi_1$ and $M \models_{h \to s'} \psi_2$;*

DPL-$\vee$: *ϕ is of the form $\psi_1 \vee \psi_2$, $s = s'$ and there exists a h such that $M \models_{s \to h} \psi_1$ or $M \models_{s \to h} \psi_2$;*

DPL-$\to$: *ϕ is of the form $\psi_1 \to \psi_2$, $s = s'$ and for all h it holds that*

$$M \models_{s \to h} \psi_1 \Rightarrow \exists h' \text{ s.t. } M \models_{h \to h'} \psi_2;$$

DPL-$\exists$: *ϕ is of the form $\exists x\psi$ and there exists an element $m \in \mathtt{Dom}(M)$ such that $M \models_{s[m/x] \to s'} \psi$;*

DPL-$\forall$: *ϕ is of the form $\forall x\psi$, $s = s'$ and for all elements $m \in \mathtt{Dom}(M)$ there exists a h such that $M \models_{s[m/x] \to h} \psi$.*

A formula ϕ is satisfied *by an assignment s if and only if there exists an assignment s' such that $M \models_{s \to s'} \phi$; in this case, we will write $M \models_s \phi$.*

We will discuss neither the formal properties of this formalism nor its linguistic applications here. All that is relevant for our purposes is that, according to it, formulas are interpreted as *transitions* from assignments to assignments, and

furthermore that the rule for conjunction allows us to bind occurrences of a variable of the second conjunct to quantifiers occurring in the first one.[7]

The similarity between this semantics and our semantics for transition terms should be evident. Hence, it seems natural to ask whether we can adopt, for a suitable variant of Dependence Logic, the following variant of Groenendijk and Stokhof's motto:

> *The meaning of a formula does not lie in its satisfaction conditions, but rather in the team transitions it allows.*

From this point of view, *transition terms* are the fundamental objects of our syntax, and formulas can be removed altogether from the language - although, of course, the tests corresponding to literals and dependence formulas should still be available. As in Groenendijk and Stokhof's logic, satisfaction becomes then a derived concept: in brief, a team X can be said to satisfy a term τ if and only if there exists a Y such that τ allows the transition from X to Y, or, in other words, if and only if *some* set of non-losing outcomes can be reached from set the initial positions X in the game corresponding to τ.

In the next section, we will make use of these intuitions to develop another, terser version of Dependence Logic; and finally, in Subsection 3.3 we will come full circle by showing how the semantics of this logic can be interpreted in terms of *reachability conditions* in a suitable variant of the Game Theoretic Semantics for standard Dependence Logic [16].

3.2 Dynamic Dependence Logic

We will now develop a formula-free variant of Transition Dependence Logic, along the lines of Groenendijk and Stockhof's Dynamic Predicate Logic.

Definition 3.2 (Dynamic Dependence Logic - Syntax) *Let Σ be a first-order signature. The set of all* transition formulas *of Dynamic Dependence Logic over Σ is given by the rules*

$$\tau ::= R\vec{t} \mid \neg R\vec{t} \mid \ =\!(t_1 \ldots t_n) \mid \exists v \mid \forall v \mid \tau \otimes \tau \mid \tau \cap \tau \mid \tau ; \tau$$

where, as usual, R ranges over all relation symbols of our signature, $\vec{t}$ ranges over all tuples of terms of the required lengths, n ranges over $\mathbb{N}$, $t_1 \ldots t_n$ range over all terms, and v ranges over $\mathtt{Var}$.

The semantical rules associated to this language are precisely as one would expect:

Definition 3.3 (Dynamic Dependence Logic - Semantics) *Let M be a first-order model, let τ be a transition formula of Dynamic Dependence Logic over the*

[7]For example, consider the formula $(\exists x Px) \wedge Qx$: by the rules given, it is easy to see that $M \models_s (\exists x Px) \wedge Qx$ if and only if $P^M \cap Q^M \neq \emptyset$, that is, if and only if $M \models_s \exists x(Px \wedge Qx)$, differently from the case of Tarski's semantics.

signature of M, and let X and Y be two teams over M with domain $\mathtt{Var}$*. Then we say that τ* allows *the transition $X \to Y$ in M, and we write $M \models_{X \to Y} \tau$, if and only if*

DDL-lit: *τ is a first-order literal, $M \models_s \tau$ in the usual first-order sense for all $s \in X$, and $X \subseteq Y$;*

DDL-dep: *τ is a dependence atom $=(t_1 \ldots t_n)$, $X \subseteq Y$, and any two assignments $s, s' \in X$ which coincide over $t_1 \ldots t_{n-1}$ also coincide over t_n;*

DDL-$\exists$: *τ is of the form $\exists v$ for some $v \in \mathtt{Var}$, and $X[F/v] \subseteq Y$ for some $F : X \to \mathtt{Dom}(M)$;*

DDL-$\forall$: *τ is of the form $\forall v$ for some $v \in \mathtt{Var}$, and $X[M/v] \subseteq Y$;*

DDL-$\otimes$: *τ is of the form $\tau_1 \otimes \tau_2$ and $X = X_1 \cup X_2$ for two teams X_1 and X_2 such that $M \models_{X_1 \to Y} \tau_1$ and $M \models_{X_2 \to Y} \tau_2$;*

DDL-$\cap$: *τ is of the form $\tau_1 \cap \tau_2$, $M \models_{X \to Y} \tau_1$ and $M \models_{X \to Y} \tau_2$;*

DDL-concat: *τ is of the form $\tau_1; \tau_2$, and there exists a Z such that $M \models_{X \to Z} \tau_1$ and $M \models_{Z \to Y} \tau_2$.*

A formula τ is said to be satisfied *by a team X in a model M if and only if there exists a Y such that $M \models_{X \to Y} \tau$; and if this is the case, we will write $M \models_X \tau$.*

It is not difficult to see that Dynamic Dependence Logic is equivalent to Transition Dependence Logic (and, therefore, to Dependence Logic).

Proposition 3.1 *Let ϕ be a Dependence Logic formula. Then there exists a term ϕ' of Dynamic Dependence Logic which is equivalent to it, in the sense that*

$$M \models_X \phi \Leftrightarrow M \models_X \phi' \Leftrightarrow \exists Y \text{ s.t. } M \models_{X \to Y} \phi'$$

for all suitable teams X and models M

Proof:
We build ϕ' by structural induction:

1. If ϕ is a literal or a dependence atom then $\phi' = \phi$;
2. If ϕ is $\psi_1 \vee \psi_2$ then $\phi' = \psi'_1 \otimes \psi'_2$;
3. If ϕ is $\psi_1 \wedge \psi_2$ then $\phi' = \psi'_1 \cap \psi'_2$;
4. If ϕ is $\exists x \psi$ then $\phi' = \exists x; \psi'$;
5. If ϕ is $\forall x \psi$ then $\phi' = \forall x; \psi'$.

□

Proposition 3.2 *Let τ be a Dynamic Dependence Logic term. Then there exists a Transition Dependence Logic term τ' such that*

$$M \models_{X \to Y} \tau \Leftrightarrow M \models_{X \to Y} \tau'$$

for all suitable X, Y and M, and such that hence

$$M \models_X \tau \Leftrightarrow M \models_X \langle \tau' \rangle \top.$$

Proof:
Build τ' by structural induction:

1. If τ is a literal or dependence atom then $\tau' = \tau?$;
2. If τ is of the form $\exists v$ or $\forall v$ then $\tau' = \tau$;
3. If τ is of the form $\tau_1 \otimes \tau_2$ then $\tau' = \tau_1' \otimes \tau_2'$;
4. If τ is of the form $\tau_1 \cap \tau_2$ then $\tau' = \tau_1' \cap \tau_2'$;
5. If τ is of the form $\tau_1; \tau_2$ then $\tau' = \tau_1'; \tau_2'$.

□

Corollary 3.4 *Dynamic Dependence Logic is equivalent to Transition Dependence Logic and to Dependence Logic*

Proof:
Follows from the two previous results and from the equivalence between Dependence Logic and Transition Dependence Logic.
□

3.3 Game Theoretic Semantics for Dynamic Dependence Logic

Apart from the team semantics (also called Hodges semantics, after its creator Wilfrid Hodges [11]) presented in Definition 1.4, Dependence Logic also has a *game theoretic semantics*, which - like the game theoretic semantics for Independence Friendly Logic - is an imperfect-information variant of the usual Game Theoretic Semantics for First Order Logic [10].

In this subsection, we will develop an analogous semantics for the case of Dynamic Dependence Logic and show that it is equivalent to the team transition semantics of Definition 3.3.

Definition 3.5 (Active Player in a Formula) *Let τ be any Dynamic Dependence Logic formula. Then $\mathtt{Player}(\tau) \in \{\mathtt{E}, \mathtt{A}\}$ is defined as follows:*

1. *If τ is a first-order literal or a dependence atom,* $\texttt{Player}(\tau) = \texttt{E}$*;*

2. *If τ is of the form $\tau_1 \otimes \tau_2$ or $\exists v$ then* $\texttt{Player}(\tau) = \texttt{E}$*;*

3. *If τ is of the form $\tau_1 \cap \tau_2$ or $\forall v$ then* $\texttt{Player}(\tau) = \texttt{A}$*;*

4. *If τ is of the form $\tau_1; \tau_2$ then* $\texttt{Player}(\tau) = \texttt{Player}(\tau_1)$.

Positions of our game will be pairs (τ, s), where τ is a transition term and s is an assignment. The *successors* of a given position are defined as follows:

Definition 3.6 *Let M be a first order model, let τ be a transition term and let s be an assignment over M. Then the set $\texttt{Succ}_M(\tau, s)$ of the* successors *of the position (τ, s) is defined as follows:*

1. *If τ is a first order literal ϕ then*

$$\texttt{Succ}_M(\tau, s) = \begin{cases} \{(\lambda, s)\} & \textit{if } M \models_s \alpha \textit{ in First Order Logic}; \\ \emptyset & \textit{otherwise} \end{cases}$$

   where λ stands for the empty string;

2. *If τ is a dependence atom then $\texttt{Succ}_M(\tau, s) = \{(\lambda, s)\}$;*

3. *If τ is of the form $\exists v$ or $\forall v$ then $\texttt{Succ}(\tau, s) = \{(\lambda, s[m/v]) : m \in \texttt{Dom}(M)\}$;*

4. *If τ is of the form $\tau_1 \otimes \tau_2$ or $\tau_1 \cap \tau_2$ then $\texttt{Succ}_M(\tau, s) = \{(\tau_1, s), (\tau_2, s)\}$;*

5. *If τ is of the form $\tau_1; \tau_2$ then*

$$\texttt{Succ}_M(\tau, s) = \{(\tau'; \tau_2, s') : (\tau', s') \in \texttt{Succ}_M(\tau_1, s)\}$$

   where, with an abuse of notation, we assume that $\lambda; \tau_2$ is equal to τ_2.

We can now define formally the semantic games associated to Dynamic Dependence Logic formulas:

Definition 3.7 ($G^M_{X \to Y}(\tau)$) *Let M be a first-order model, let τ be a Dynamic Dependence Logic formula, and let X and Y be teams. Then the game $G^M_{X \to Y}(\tau)$ is defined as follows:*

- *The set **I** of the* initial positions *of the game is $\{(\tau, s) : s \in X\}$;*
- *The set **W** of the* winning positions *of the game is $\{(\lambda, s) : s \in Y\}$;*
- *For any position (τ', s'), the* active player *is $\texttt{Player}(\tau')$ and the* set of successors *is $\texttt{Succ}_M(\tau', s')$.*

Definition 3.8 (Play) *Let $G^M_{X \to Y}(\tau)$ be as in the above definition. Then a* play *of this game is a finite sequence $\vec{p} = p_1 \dots p_n$ of positions of the game such that*

1. *$p_1 \in$ **I** is an* initial position *of the game;*

2. *For every $i \in 1 \ldots n-1$, $p_{i+1} \in \texttt{Succ}_M(p_1)$.*

If furthermore $\texttt{Succ}_M(p_n) = \emptyset$, we say that $\vec{p}$ is complete*; and if $p_n \in \boldsymbol{W}$ is a* winning position*, we say that $\vec{p}$ is* winning.

So far, we did not deal with the satisfaction conditions of dependence atoms at all. Similarly to the case of Dependence Logic [16], such conditions are made to correspond as *uniformity conditions* over sets of plays:

Definition 3.9 (Uniform set of plays) *Let $G^M_{X \to Y}(\tau)$ be a game, and let P be a set of plays in it. Then P is* uniform *if and only if for all $\vec{p}, \vec{q} \in P$ and for all i, j, such that $p_i = (=(t_1 \ldots t_n); \tau, s)$ and $q_j = (=(t_1 \ldots t_n); \tau, s')$ for the same instance of the dependence atom $=(t_1 \ldots t_n)$ it holds that*

$$(t_1 \ldots t_{n-1})\langle s \rangle = (t_1 \ldots t_{n-1})\langle s' \rangle \Rightarrow t_n\langle s \rangle = t_n\langle s' \rangle$$

where, with an abuse of notation, we identify $=(t_1 \ldots t_n); \lambda$ with $=(t_1 \ldots t_n)$.

We will only consider *positional strategies*, that is, strategies that depend only on the current position.

Definition 3.10 (Local Strategy) *Let $G^M_{X \to Y}(\tau)$ be as above, and let τ' be any expression such that (τ', s') is a possible position of the game for some s'. Then a* local strategy *for τ' is a function $f_{\tau'}$ sending each s' into a $(\tau'', s'') \in \texttt{Succ}_M(\tau', s')$.*

Definition 3.11 (Plays Following a Local Strategy) *Let $G^M_{X \to Y}(\tau)$ be as above, let $\vec{p} = p_1 \ldots p_n$ be a play in it, and let $f_{\tau'}$ be a local strategy for some τ'. Then $\vec{p}$ is said to* follow *$f_{\tau'}$ if and only if for all $i \in 1 \ldots n-1$ and all s',*

$$p_i = (\tau', s') \Rightarrow p_{i+1} = f_{\tau'}(s').$$

Definition 3.12 (Global Strategy) *Let $G^M_{X \to Y}(\tau)$ be as above. Then a* global strategy *(for $\texttt{E}$) in this game is a function f associating to each expression τ' occurring in some nonterminal position of the game and such that $\texttt{Player}(\tau') = \texttt{E}$ with some local strategy $f_{\tau'}$ for τ'.*

Definition 3.13 (Plays following a global strategy) *A play $\vec{p}$ of a game $G^M_{X \to Y}(\tau)$ is said to* follow *a global strategy f if and only if it follows $f_{\tau'}$ for all τ'.*

Definition 3.14 (Winning Strategy) *A global strategy f for a game $G^M_{X \to Y}(\tau)$ is said to be* winning *if and only if all complete plays which follow f are winning.*

Definition 3.15 (Uniform Strategy) *A global strategy f for a game $G^M_{X \to Y}(\tau)$ is said to be* uniform *if and only if the set of all complete plays which follow f respects the uniformity condition of Definition 3.9.*

The following result then connects the game theoretic semantics we just defined and the team transition semantics for Dynamic Dependence Logic:

Theorem 3.16 *Let M be a first-order model, let X and Y be teams, and let τ be any Dynamic Dependence Logic transition term. Then $M \models_{X \to Y} \tau$ if and only if the existential player* $\mathtt{E}$ *has a uniform winning strategy for $G^M_{X \to Y}(\tau)$.*

Proof:
The proof is by structural induction on τ:

1. If τ is a first-order literal and $M \models_{X \to Y} \tau$, then $X \subseteq Y$ and $M \models_s \tau$ in the usual first-order sense for all $s \in X$. Then there exists only one strategy f for $\mathtt{E}$ in $G^M_{X \to Y}(\tau)$, and for this strategy we have that $f_\tau(s) = (\lambda, s)$ for all $s \in X$. Since $X \subseteq Y$, this strategy is winning; and furthermore, it is trivially uniform. Hence, $\mathtt{E}$ has a uniform winning strategy in $G^M_{X \to Y}(\tau)$.

   Conversely, suppose that $\mathtt{E}$ has an uniform winning strategy f in $G^M_{X \to Y}(\tau)$. If $M \not\models_s \tau$ for some $s \in X$, then the position (τ, s) is terminal in this game and it is not winning, which contradicts our hypothesis. Hence $M \models_s \tau$ for all $s \in X$, and furthermore - since $f_\tau(s) = (\lambda, s)$ for all $s \in X$ - we have that $X \subseteq Y$. Thus, $M \models_{X \to Y} \tau$, as required.

2. If τ is a dependence atom $=\!(t_1 \dots t_n)$ and $M \models_{X \to Y} \tau$, then $X \subseteq Y$ and X satisfies the dependency condition associated with the atom. But then the only strategy f available to player $\mathtt{E}$ is winning, as the set of its terminal position is $\{(\lambda, s) : s \in X\}$ and $X \subseteq Y$, and it is also uniform, since the set of all possible plays of the game is $\{(=\!(t_1 \dots t_n, s))(\lambda, s) : s \in X\}$.

   Conversely, suppose that the only strategy f available to $\mathtt{E}$ in $G^M_{X \to Y}(\tau)$ is uniform and winning. Since it is uniform, it follows at once that X satisfies the dependency condition; and since it is winning and the set of all terminal positions is $\{(\lambda, s) : s \in X\}$, we have that $X \subseteq Y$, and hence that $M \models_{X \to Y} \tau$.

3. If τ is $\exists v$ for some variable v and $M \models_{X \to Y} \tau$, then $X[F/v] \subseteq Y$ for some $F : X \to \mathtt{Dom}(M)$. Now let f be the strategy for $\mathtt{E}$ in $G^M_{X \to Y}(\tau)$ such that $f_\tau(s) = s[F(s)/v]$: this strategy is uniform, and the set of its terminal positions is $\{(\lambda, s[F(s)/v]) : s \in X\} = \{(\lambda, s') : s' \in X[F/v]\}$. Hence, f_τ is also winning, as required.

   Conversely, let f be any uniform winning strategy for $\mathtt{E}$ in $G^M_{X \to Y}(\tau)$, and define $F : X \to \mathtt{Dom}(M)$ so that

   $$f_\tau(s) = (\lambda, s[F(s)/v])$$

   for all $s \in X$.

   Since f is winning, $f_\tau(s)$ is a winning position for all $s \in X$, and hence $X[F/v] = \{s[F(s)/v] : s \in X\} \subseteq Y$. Hence, $M \models_{X \to Y} \tau$, as required.

4. If τ is $\forall v$ for some variable v and $M \models_{X \to Y} \tau$, then $X[M/v] \subseteq Y$. There exists only one (trivial, and trivially uniform) strategy for $\mathtt{E}$ in the game $G^M_{X \to Y}(\tau)$, as the universal player $\mathtt{A}$ moves in all non-terminal positions; and the set of all possible outcomes of the game for all the initial positions

is $\{(\lambda, s[m/v]) : s \in X, m \in \texttt{Dom}(M)\} = \{(\lambda, s') : s' \in X[M/v]\}$. Hence this strategy is winning, as required.

Conversely, suppose that the unique strategy for $\texttt{E}$ in $G^M_{X\to Y}(\tau)$ is winning for this game. Then, as the set of all possible outcomes is $\{(\lambda, s') : s' \in X[M/v]\}$, we have that $X[M/v] \subseteq Y$, and hence that $M \models_{X\to Y} \tau$.

5. If τ is $\tau_1 \otimes \tau_2$ for two transition terms τ_1 and τ_2 and $M \models_{X\to Y} \tau_1 \otimes \tau_2$, then $X = X_1 \cup X_2$ for two X_1 and X_2 such that $M \models_{X_1\to Y} \tau_1$ and $M \models_{X_2\to Y} \tau_2$. By induction hypothesis, this implies that there exist two strategies f_1 and f_2 for $\texttt{E}$ which are uniform and winning in $G^M_{X_1\to Y}(\tau_1)$ and $G^M_{X_2\to Y}(\tau_2)$ respectively. Now define a strategy f for $\texttt{E}$ in $G^M_{X\to Y}(\tau_1 \otimes \tau_2)$ as follows:

   - If τ' is part of τ_1 then $f_{\tau'} = (f_1)_{\tau'}$;
   - If τ' is part of τ_2 then $f_{\tau'} = (f_2)_{\tau'}$;
   - If τ' is $\tau_1 \otimes \tau_2$ then $f_{\tau'}(s) = \begin{cases} (\tau_1, s) & \text{if } s \in X_1; \\ (\tau_2, s) & \text{if } s \in X_2 \backslash X_1. \end{cases}$

   This strategy is clearly uniform, as f_1 and f_2 are uniform. Furthermore, it is winning: indeed, any play of $G^M_{X\to Y}(\tau_1 \otimes \tau_2)$ in which $\texttt{E}$ follows it strictly contains a play of $G^M_{X_1\to Y}(\tau)$ in which $\texttt{E}$ follows f_1 or a play of $G^M_{X_2\to Y}(\tau)$ in which $\texttt{E}$ follows f_2, and in either case the game ends in a winning position in Y.

   Conversely, suppose that f is a uniform winning strategy for $\texttt{E}$ in $G^M_{X\to Y}(\tau)$. Now let $X_1 = \{s \in X : f_\tau(s) = (\tau_1, s)\}$, let $X_2 = \{s \in X : f_\tau(s) = (\tau_2, s)\}$, and let f_1 and f_2 be the restrictions of f to the subgames corresponding to τ_1 and τ_2 respectively. Then f_1 and f_2 are uniform and winning for $G^M_{X_1\to Y}(\tau_1)$ and $G^M_{X_2\to Y}(\tau_2)$ respectively, and hence by induction hypothesis $M \models_{X_1\to Y} \tau_1$ and $M \models_{X_2\to Y} \tau_2$. But $X = X_1 \cup X_2$, and hence this implies that $M \models_{X\to Y} \tau$.

6. If τ is $\tau_1 \cap \tau_2$ for some τ_1 and τ_2 and $M \models_{X\to Y} \tau_1 \cap \tau_2$, then $M \models_{X\to Y} \tau_1$ and $M \models_{X\to Y} \tau_2$. By induction hypothesis, this implies that $\texttt{E}$ has two uniform winning strategies f_1 and f_2 for $G^M_{X\to Y}(\tau_1)$ and $G^M_{X\to Y}(\tau_2)$ respectively. Now let f be the strategy for $G^M_{X\to Y}(\tau_1 \cap \tau_2)$ which behaves like f_1 over the subgame corresponding to τ_1 and like f_2 over the subgame corresponding to τ_2 (it is not up to $\texttt{E}$ to choose the successors of the initial positions $(\tau_1 \cap \tau_2, s)$, so she needs not specify a strategy for those). This strategy is winning and uniform, as required, because τ_1 and τ_2 are so.

   Conversely, suppose that $\texttt{E}$ has a uniform winning strategy f for $G^M_{X\to Y}(\tau_1 \cap \tau_2)$. Since the opponent $\texttt{A}$ chooses the successor of the initial positions $\{(\tau_1 \cap \tau_2, s) : s \in X\}$, any element of $\{(\tau_1, s) : s \in X\}$ and of $\{(\tau_2, s) : s \in X\}$ can occur as part of a play in which $\texttt{E}$ follows f. Now, let f_1 and f_2 be the restrictions of f to the subgames corresponding to τ_1 and τ_2 respectively: then f_1 and f_2 are uniform, because f is so, and they are

winning for $G^M_{X\to Y}(\tau_1)$ and $G^M_{X\to Y}(\tau_2)$ respectively, because every play of these games in which E follows f_1 (resp f_2) starting from a position (τ_1, s) (resp. (τ_2, s)) for $s \in X$ can be transformed into a play of $G^M_{X\to Y}(\tau_1 \cap \tau_2)$ in which E follows f simply by appending the initial position $(\tau_1 \cap \tau_2, s)$ at the beginning.

7. If τ is $\tau_1; \tau_2$ for some τ_1 and τ_2 and $M \models_{X\to Y} \tau_1; \tau_2$, then there exists a Z such that $M \models_{X\to Z} \tau_1$ and $M \models_{Z\to Y} \tau_2$. By induction hypothesis, this implies that there exist two strategies f_1 and f_2 which are winning for E in $G^M_{X\to Z}(\tau_1)$ and in $G^M_{Z\to Y}(\tau_2)$ respectively. Now define a strategy f for E in $G^M_{X\to Y}(\tau_1; \tau_2)$ as

   - If $(f_1)_{\tau'}(s') = (\tau'', s'')$ then $f_{\tau';\tau_2}(s') = (\tau''; \tau_2, s'')$;
   - If τ' is part of τ_2 then $f_{\tau_2} = (f_2)_{\tau_2}$.

   We need to prove that this f is uniform and winning. Now, let us consider the set of all plays in which E follows f: it is easy to see that they will be played exactly as a game of $G^M_{X\to Z}(\tau_1)$ until a position of the form (τ_2, s') is reached for some $s' \in Z$, and then they will be played exactly as a game of $G^M_{Z\to Y}(\tau_2)$ until a position of the form (λ, s'') is reached for some $s'' \in Y$. Hence, the strategy is winning, as it will always end in a winning position for Y, and it is uniform, because any violation of uniformity would also be a violation for f_1 or f_2.

   Conversely, let f be a uniform winning strategy for E in $G^M_{X\to Y}(\tau_1; \tau_2)$, and let Z be the set of all assignments s such that the position (τ_2, s) occurs as part of some play of $G^M_{X\to Y}(\tau_1; \tau_2)$ in which E follows f. Furthermore, let the two strategies f_1 and f_2 for E in $G^M_{X\to Z}(\tau_1)$ and $G^M_{Z\to Y}(\tau_2)$ respectively be defined as

   - If $f_{\tau';\tau_2}(s') = (\tau''; \tau_2, s'')$ then $(f_1)_{\tau'}(s) = (\tau'', s'')$;
   - If τ' is part of τ_2 then $(f_2)_{\tau_2} = f_{\tau_2}$.

   By construction and definition, it follows at once that τ_1 and τ_2 are uniform winning strategies for $G^M_{X\to Z}(\tau_1)$ and $G^M_{Z\to Y}(\tau_2)$ respectively. By induction hypothesis, this implies that $M \models_{X\to Z} \tau_1$ and that $M \models_{Z\to Y} \tau_2$, and finally that $M \models_{X\to Y} \tau_1; \tau_2$.

$\square$

Theorem 3.16 shows that Dynamic Dependence Logic can be interpreted in terms of *reachability*: $M \models_{X\to Y} \tau$ if and only if, in the game corresponding to τ, the existential player can guarantee that the final assignment will be in Y whenever the initial assignment is in X. This corresponds exactly to the intuitions behind the notion of *transition system* which we introduced in Subsection 2.1, and further confirms that Dependence Logic and its variants are suitable frameworks for exploring decision-theoretic reasoning under imperfect information in a first-order setting.

4 Acknowledgements

The author wishes to thank Johan van Benthem and Jouko Väänänen for a number of useful suggestions and insights.

References

[1] P. Dekker. A guide to dynamic semantics. *ILLC Prepublication Series*, (PP-2008-42), 2008.

[2] Fredrik Engström. Generalized quantifiers in dependence logic. *Journal of Logic, Language and Information (to appear)*, 2011.

[3] Pietro Galliani. Multivalued dependence logic and independence logic. In *Non-classical Modal and Predicate Logics*, 2011.

[4] Pietro Galliani. Inclusion and exclusion dependencies in team semantics: On some logics of imperfect information. *Annals of Pure and Applied Logic*, 163(1):68 – 84, 2012.

[5] Erich Grädel and Jouko Väänänen. Dependence and Independence. *Studia Logica (to appear)*, 2010.

[6] J. Groenendijk and M. Stokhof. Dynamic Predicate Logic. *Linguistics and Philosophy*, 14(1):39–100, 1991.

[7] L. Henkin. Some Remarks on Infinitely Long Formulas. In *Infinitistic Methods. Proc. Symposium on Foundations of Mathematics*, pages 167–183. Pergamon Press, 1961.

[8] Jaakko Hintikka. *The Principles of Mathematics Revisited*. Cambridge University Press, 1996.

[9] Jaakko Hintikka and Gabriel Sandu. Informational independence as a semantic phenomenon. In J.E Fenstad, I.T Frolov, and R. Hilpinen, editors, *Logic, methodology and philosophy of science*, pages 571–589. Elsevier, 1989.

[10] Jaakko Hintikka and Gabriel Sandu. Game-Theoretical Semantics. In Johan van Benthem and Alice T. Meulen, editors, *Handbook of Logic and Language*, pages 361–410. Elsevier, 1997.

[11] Wilfrid Hodges. Compositional Semantics for a Language of Imperfect Information. *Journal of the Interest Group in Pure and Applied Logics*, 5 (4):539–563, 1997.

[12] Juha Kontinen and Ville Nurmi. Team logic and second-order logic. In Hiroakira Ono, Makoto Kanazawa, and Ruy de Queiroz, editors, *Logic, Language, Information and Computation*, volume 5514 of *Lecture Notes in Computer Science*, pages 230–241. Springer Berlin / Heidelberg, 2009.

[13] Juha Kontinen and Jouko Väänänen. On definability in dependence logic. *Journal of Logic, Language and Information*, 3(18):317–332, 2009.

[14] Allen L. Mann, Gabriel Sandu, and Merlijn Sevenster. *Independence-Friendly Logic: A Game-Theoretic Approach*. Cambridge University Press, 2011.

[15] Rohit Parikh. The logic of games and its applications. In *Selected papers of the international conference on "foundations of computation theory" on Topics in the theory of computation*, pages 111–139, New York, NY, USA, 1985. Elsevier North-Holland, Inc.

[16] Jouko Väänänen. *Dependence Logic*. Cambridge University Press, 2007.

[17] Jouko Väänänen. Team Logic. In J. van Benthem, D. Gabbay, and B. Löwe, editors, *Interactive Logic. Selected Papers from the 7th Augustus de Morgan Workshop*, pages 281–302. Amsterdam University Press, 2007.

[18] Johan van Benthem. Logic games are complete for game logics. *Studia Logica*, 75:183–203, 2003.

[19] Johan van Benthem, Sujata Ghosh, and Fenrong Liu. Modelling simultaneous games in dynamic logic. *Synthese*, 165:247–268, 2008.